НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Теплоенергетичний факультет
Кафедра автоматизації проектування енергетичних процесів і систем

До захисту допущено:
Завідувач кафедри
_____ Олександр Коваль
«___»_____2020 р.

Дипломна робота

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Програмне забезпечення веб-технологій та мобільних пристроїв»

спеціальності 121 «Інженерія програмного забезпечення»

на тему: «Програмний комплекс для контролю

доступу у загальну систему безпеки

НТУУ «КПІ ім. Ігоря Сікорського»

Виконав (-ла):
студент (-ка) IV курсу, групи ТІ-62
Велігоненко Сегій Олександрович _____

Керівник:
Коваль Олександр Васильович _____

Консультант з назва розділу:
Посада, науковий ступінь, вчене звання,
Прізвище, ім’я, по батькові _____

Рецензент:
Посада, науковий ступінь, вчене звання,
Прізвище, ім’я, по батькові _____

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент (-ка) _____

Київ – 2020 року

Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти перший рівень

Напрям підготовки: 121 Інженерія програмного забезпечення

Спеціалізація: Програмне забезпечення веб-технологій та мобільних пристроїв

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____Олександр Коваль
(підпис)
”___”_____2020р.

ЗАВДАННЯ
на дипломну роботу студенту

_____
(прізвище, ім’я, по батькові)

1. Тема роботи Програмний комплекс для контролю доступу у загальну систему безпеки НТУУ «КПІ ім. Ігоря Сікорського»
_____
керівник роботи___Коваль Олександр Васильович _____
(прізвище, ім’я, по батькові науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу від ”25” травня 2020р. **№ 1168-с**

2. Строк подання студентом роботи _____
3. Вихідні дані до роботи ____Технічне завдання_____
_____
_____

4.Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)____Актуальність, мета роботи, постановка задачі, технології рохробки_____
_____
_____
_____
_____
_____
_____

_____

5. Перелік ілюстративного матеріалу

_____

_____

_____

_____

_____

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання ”___”_____201___ р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітки
1.	Затвердження теми роботи	25.02.2020	
2.	Вивчення та аналіз задачі	6.03.2020	
3.	Розробка архітектури та загальної структури системи	11.03.2020	
4.	Розробка структур окремих підсистем	28.03.2020	
5.	Програмна реалізація системи	15.04.2020	
6.	Оформлення пояснювальної записки	29.04.2020	
7.	Захист програмного продукту	18.05.2020	
8.	Передзахист	11.06.2020	
9.	Захист	15.06.2020	

Студент _____ _____
(підпис) (прізвище та ініціали,)

Керівник роботи _____ _____
(підпис) (прізвище та ініціали,)

1. Анотація

Дана робота присвячена розробці платформи для контролю доступу.

Метою роботи є аналіз застосунків-конкурентів й розробка з подальшим впровадженням власної системи, в якій відсутні недоліки аналогів.

В ході роботи було вирішено наступні задачі:

1. Провести аналіз застосунків-конкурентів.
2. Визначити засоби розробки для створення аналітичної платформи.
3. Здійснити проєктний аналіз і розробити каркас застосунку.
4. Забезпечити безперебійну роботу платформи.

В роботі було використано платформу JVM, для якої розробка здійснювалась засобами мови програмування Java.

Ключові слова: безпека, контроль доступу, база даних.

Abstract

This thesis is devoted to development of an access control platform.

The purpose of the work is to examine existing analogues of such system, design and develop own solution without disadvantages of analogues.

In order to achieve this goal several tasks were solved:

1. Analyze existing analogues.
2. Define tools for development of an analytical platform.
3. Design and implement application framework.
4. Ensure uninterrupted operation of the platform.

The JVM platform was used in the work, for which the development was carried out by means of Java programming language.

ЗМІСТ

ВСТУП 7
1 ПОСТАНОВКА ЗАДАЧІ 8
2 АНАЛІЗ СИСТЕМ УПРАВЛІННЯ ТА КОНТРОЛЮ ДОСТУПУ 9
3 ЗАСОБИ РОЗРОБКИ 12
3.1 Середовище розробки Intellij Idea 12
3.2 Візуальний інтерфейс HTML, CSS, JavaScript 12
3.3 Серверна мова програмування 16
4 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ 19
4.1 MVC 19
4.2 База даних MySQL 20
4.3 TDD 21
4.3 Шифрування 22
5 РОБОТА КОРИСТУВАЧА З ПРОГРАМНОЮ СИСТЕМОЮ 24
5.1 Системні вимоги 24
5.2 Робота аналітика з програмним продуктом 24
6 ВИСНОВКИ 25
ДОДАТОК 1 26

ВСТУП

В наш час до захисту особистої інформації відносяться ретельно. У першу чергу це пояснюється можливостю використати особисті дані людини проти її волі задля власної вигоди. Саме для цього існують системи контролю доступу, які унеможливлюють доступ сторонніх осіб до персональних даних. Відповідно до рівня важливості даних існують різні методи аутентифікації користувача. Наприклад, для особистої поштової скриньки достатньо логіна і пароля, а для більш важливих даних може знадобитись двухфакторна аутентифікація або електронний ключ.

Запропонована система вирішує питання контролю доступу та моніторингу запитів до даних. Для зручного користування був розроблений веб-інтерфейс, а для забезпечення роботи системи було розроблено серверну частину на мові Java.

1 ПОСТАНОВКА ЗАДАЧІ СТВОРЕННЯ ПРОГРАМНОГО КОМПЛЕКСУ ДЛЯ КОНТРОЛЮ ДОСТУПУ У ЗАГАЛЬНУ СИСТЕМУ БЕЗПЕКИ «КПІ ім. І.СІКОРСЬКОГО»

За допомогою теоритичного матеріалу та практичного досвіду, отриманого в лабораторії, розробити програмний комплекс. Програмний продукт має бути кросплатформним, тобто повинен однаково працювати на різних платформах, що забезпечує великий обсяг аудіторії. Також система повинна надавати доступ тільки авторізованим користувачам та функціонал згідно з рівнями доступу. Система повинна бути захищенної від несанкціанового доступу, та зберігати інформацію у зашифрованому вигляді.

Після аналізу вимог для розробки комплексу програм була використана мова програмування Java. Розробка графічного інтерфейсу користувача відбулась на основі HTML5, CSS, JavaScript.

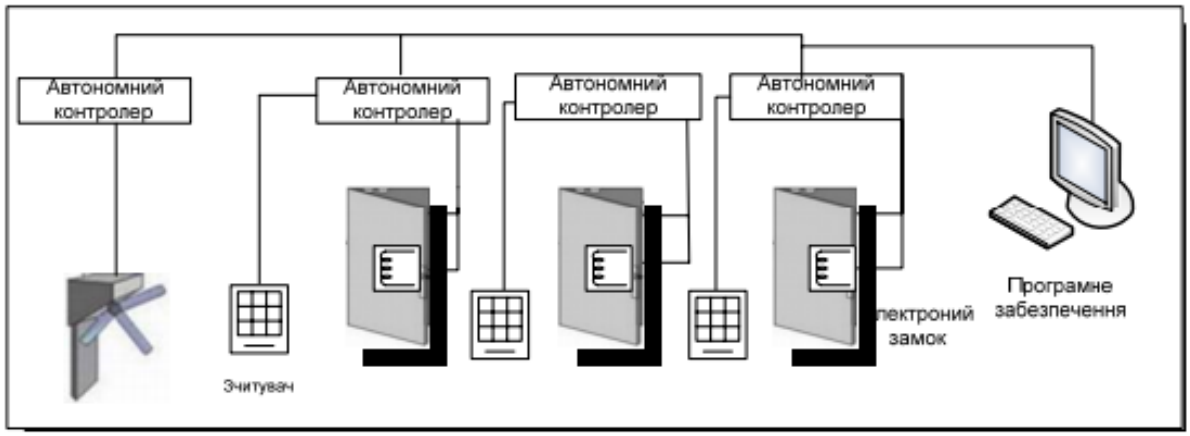
2 АНАЛІЗ СИСТЕМ УПРАВЛІННЯ ТА КОНТРОЛЮ ДОСТУПУ

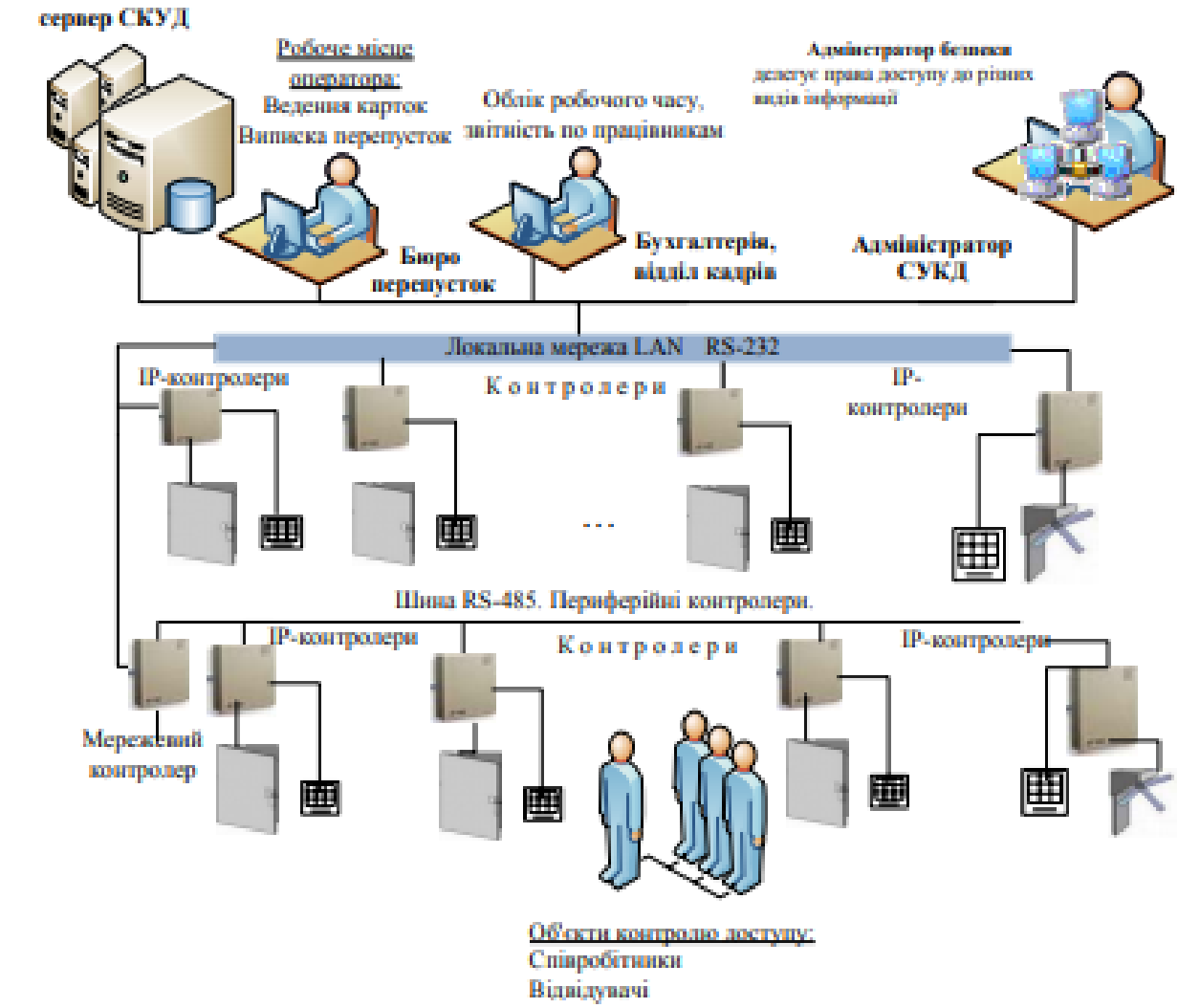
Системою контролю та управління доступу називають сукупність програмно-апаратних технічних засобів безпеки, що мають на меті: обмеження і реєстрацію входу-виходу об’єктів (людей, транспорту) на заданій території через «точки проходу. Також, СКУД використовують для збору різноманітної інформації про працівників, їх пересування територією підприємства, термінів і часу знаходження в підрозділах та ін. Сучасні СКУД складаються з таких елементів:

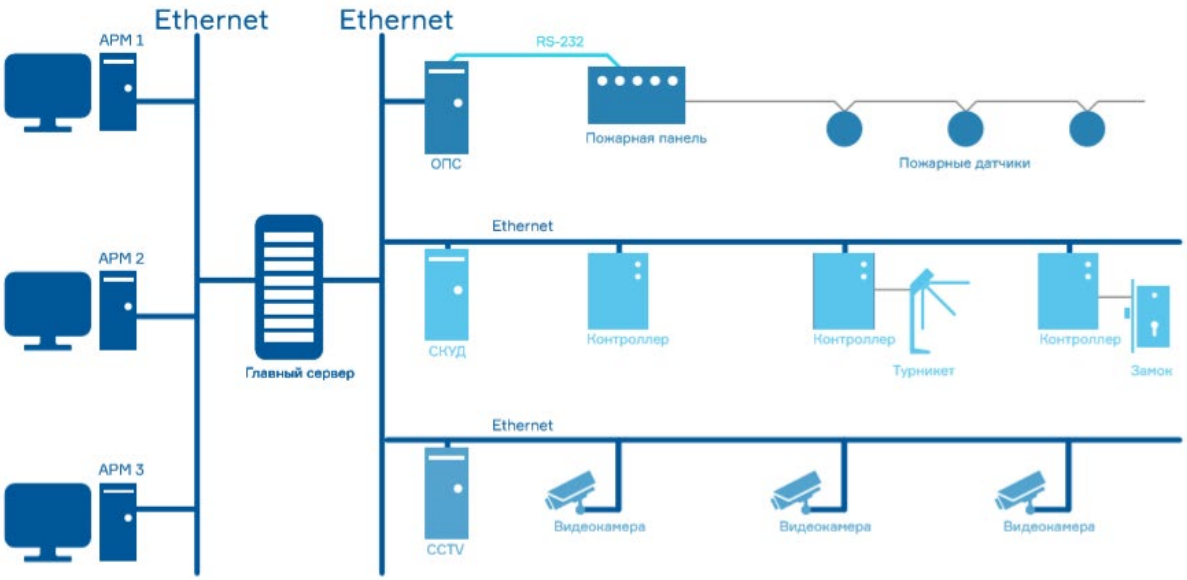
1. Ідентифікатор – пристрій, який дозволяє ідентифікувати користувача. Популярні зараз безконтактні карти, контактні карти, електронні картки. Ще в ролі ідентифікатора може виступати спеціальний код, який вводить людина під час входу в зону, що охороняється, або біометричні дані – відбитки пальців або долоні, відбиток сітківки очей, розпізнавання голосу і т. д.

2. Зчитувач – пристрій для розпізнавання і передачі даних на керуючий блок.

3. Керуючий блок – сама система. Виконує головну роль при визначенні надавання доступу до даних користувачу.







Недоліки: Залежність швидкості обробки даних від розташування сервера. Складність оновлення, бо програми розроблені на застарілих технологіях.

3 ЗАСОБИ РОЗРОБКИ

Найважливішим чинником, під час розробки програмного продукту, є вибір засобів програмної реалізації та технологій. Середовищем розробки інформаційної системи було обрано Intellij Idea Community Edition. Мова для програмування Java 8. Для створення графічного інтерфейсу системи використовувались HTML5, CSS, JavaScript.

Для роботи з базами даних використовувалась система управління базами даних MySQL, а для зручної роботи MySQL Workbench.

3.1 Середовище розробки Intellij Idea.

Intellij Idea – це зручна платформа розробки для багатьох операційних систем. Це дозволяє користувачам не витрачати свій час та ресурси на розробку одного продукту для декількох платформ, а зразу зосередитись на якості програмного продукту.

За допомогою зручного інтерфейсу розробники також мають доступ до безлічі інструментів налагодження. Більше того, Intellij Idea також підтримує тести, тобто розробники можуть імітувати різні умови для своєї програми та проаналізувати можливі проблеми, що дозволить виправити на ранній стадії. Також ця платформа підтримує багато розширень та плагінів, які можна встановити з магазину. Це дозволяю налаштувати середовище під себе та розширити функціонал середовища розробки.

3.2 Візуальний інтерфейс HTML, CSS, JavaScript

Візуальна частина представлена за допомогою HTML, CSS, JavaScript.

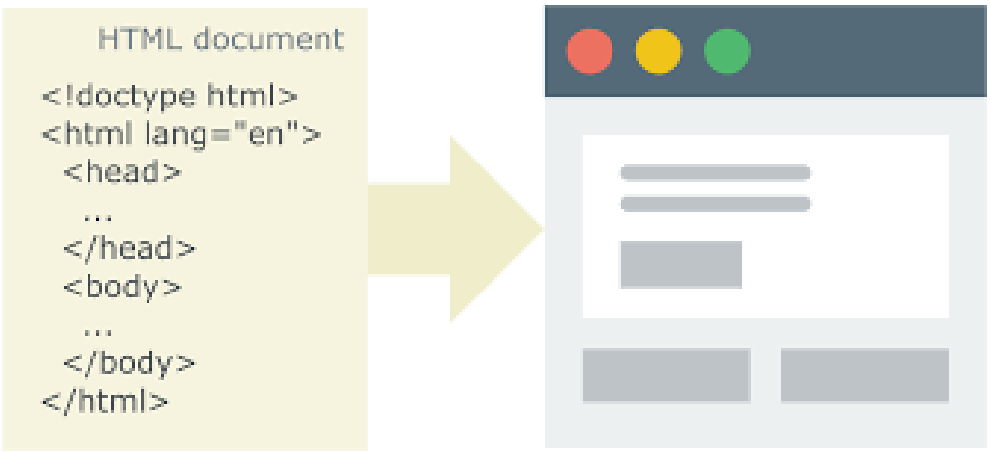
HTML використовувався для розмітки документів у Всесвітній павутині. Мова HTML інтерпретується браузерами і отриманий результат відображається на екрані монітора або мобільного пристрою у зручному для користувачу вигляді

CSS додає документу стилі, оформлення, що робить сторінку привабливою для користувачів. Саме такий підхід дозволяє представити користувачеві сторінку краще, ніж за допомогою гіпертекстової розмітки. За допомогою CSS розробник може зробити кожен елемент привабливим і зручним для користування. На сьогоднішній день всі сайти використовують цю технологію. Однією з переваг є те, що кожен стиль має своє індивідуальне правило показу, тобто дії будуть виконуватися за принципом – правило і його визначення, а також необхідні елементи (теги), до яких всі параметри будуть застосовуватися.

JavaScript відповідає за анімацію та опрацьовує дані на боці користувача. Разом ці технології створюють веб-сайт.

Мова JavaScript використовується для:

- написання сценаріїв веб-сторінок для надання їм інтерактивності;
- створення односторінкових веб-застосунків
- програмування на стороні сервера
- стаціонарних застосунків
- мобільних застосунків
- сценаріїв в прикладному ПЗ
- всередині PDF-документів тощо.



3.3 Серверна мова програмування

Програмування на боці сервера використовується у веб-програмуванні для надання індивідуальних відповіей кожному клієнту, що звертається до веб-сайту. Програмування на стороні сервера відрізняється від застосування скриптів на стороні клієнта, де вбудовані скрипти у веб-сторінку написані, наприклад, на Java-script виконуються у веб-браузері клієнта. Проте обидва технічні прийоми часто використовуються разом. Важливі та вразливі дії відбуваються на сервері для забезпечення безпеки та надійної роботи. Деякі операції можуть виконуватись на боці клієнта для поліпшення швидкості роботи веб-системи.

Серверне програмування використовується для надання користувачу загального інтерфейсу з індивідуальними даними. Такий підхід дозволяє сховати код, що забезпечує безпеку. Виконувані скрипти на боці клієнта є вразливими, бо код доступний для всіх. Мінусом такого підходу є те, що при програмуванні на стороні сервера, клієнт повинен надсилати запити через мережу до сервера, щоб отримати нову інформацію через користувацький веб-браузер. Такий підхід може уповільнити роботу веб-системи, потребує стабільного підключення до Всесвітньої мережи і залежить від її швидкості.

У якості серверної мови було обрано Java. Перевага саме цій мові була віддана через незалежність від платформи на якій виконується один і той самий код, строгу типізацію даних, об’єктно орієнтованість, зручний синтаксис, а також зручне програмування. Для цього в Java позбавили користувача можливості явного виділення пам’яті, що дуже часто приводило до помилок, які складно шукати. Багато технічної літератури, уроків, документації та розвинуте ком’юніті також є перевагою. Було прийняти рішення зупинитись на восьмій версій, бо наразі ця версія є популярною і дає багато можливостей для розробника, наприклад з’явились лямбди і стріми.

Програмування існує вже досить тривалий час і багато програмістів зустрічаються з однаковими проблемами та задачами. Для написання такого

загального коду витрачається багато ресурсів. Аби прискорити розробку і звільнити програмістів від написання загального коду було створено фреймворки. Фреймворк – це готовий до використання комплекс програмних рішень, включаючи дизайн, логіку та базову функціональність системи або підсистеми. Виходячи з цього програмний фреймворк може навіть містити в собі допоміжні програми, деякі бібліотеки коду, скрипти та загалом все, що полегшує створення та поєднання різних компонентів великого програмного забезпечення чи швидке створення готового і не обов'язково об'ємного програмного продукту. Побудова кінцевого продукту відбувається, зазвичай, на базі єдиного API. Таким чином надається змога швидко розробити застосунок і дозволяє програмісту швидко перейти до написання специфічного коду для веб-системи.

Одна з головних переваг такого підходу це те, що такі програми мають стандартну структуру. Такий код, як правило, стандартизований згідно вимог програмування та оптимізований. Фреймворки надають інтерфейси для роботи, що дозволяє відокремитись від реалізації і програмувати на рівні інтерфейсів, що є гарною практикою у світі програмування. З їх появленням стало набагато простіше створювати засоби для автоматичного створення графічних інтерфейсів, оскільки структура внутрішньої реалізації коду програми стала відома заздалегідь. Для забезпечення каркаса, зазвичай, використовують підходи об'єктно-орієнтованого програмування, наприклад, частини програми можуть успадковуватися від базових класів фреймворку. На сьогоднішній день існує багато різних фреймворків з широким функціоналом

Популярні фреймворки:

- Django
- Rails
- Laravel
- Spring
- Vue

Для більш швидкого і зручного програмування було використано фреймворк Spring Boot. Це один з найпоширеніших фреймворків для розробки. Він містить у собі багато шаблонів, що дозволяє користувачу менше витрачати часу на написання шаблонного коду. У нашому продукті більше переваги надавалось Spring Security, MVC, Testing. Spring Security відповідає за аутентифікацію користувача і доступ відповідно до прав. Spring Testing дозволяє перевірити код у різних умовах та виправити помилки. Покриття тестами також є показником якості програмного продукту. Spring MVC дозволяє швидко приступити до написання програми за шаблоном MVC

4 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

4.1 MVC

Model-View-Controller (Модель-представлення-контролер, англ. MVC) – архітектурний шаблон, який використовується під час проектування та розробки програмного забезпечення.

Цей шаблон дозволяє поділити програму на три основні частини. Модель є найголовнішою частиною і відповідає за поведінку програми та функціонал. Контролер перетворює сигнали, аналізує і відправляє на модель або на зображення. Зображення відповідає тільки за відображення даних для користувача. Саме такий підхід є зараз поширеним у світі розробки і найбільшою перевагою цього шаблону є те, що частини не залежать одна від одної і можуть легко бути заміненими на іншу реалізацію у короткий проміжок часу.

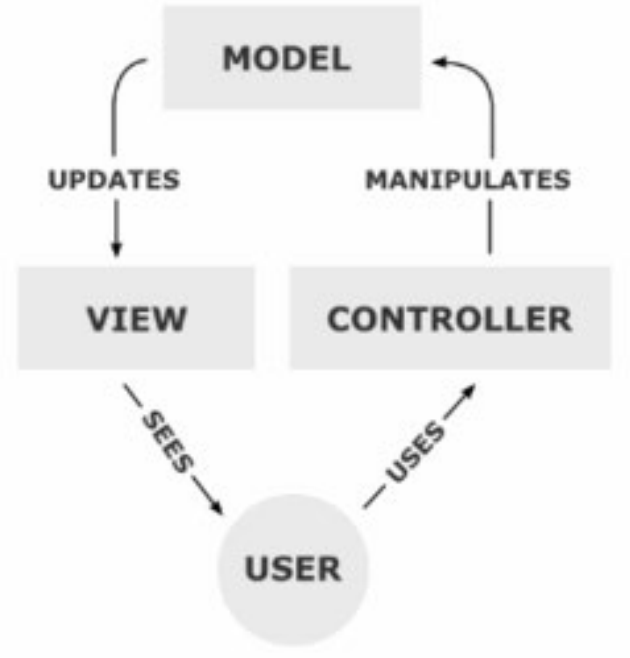


Рис. 4.1.1- шаблон MVC

4.2 База даних MySQL

Швидкість роботи програми залежить від правильно обраної і спроектованої бази даних. На ранніх стадіях, коли даних мало, швидкість може бути задовільною. Но як тільки база даних почне зростати, швидкість програмного продукту може дуже сильно зменшуватись. Перейти на іншу базу може бути дуже складним завданням, яке потребує багато часу та ресурсів. Тому важливо до написання коду проаналізувати всі вимоги а також дані, які будуть зберігатись у базі, і обрати базу даних, яка задовольняє усім потребам. Бази даних поділяються на реляційні і нереляційні. Найбільш поширеними є реляційні, бо нереляційні дуже специфічні і мають великі відмінності.

MySQL, створена в 1994 році, є однією з най поширених СУБД і з тих пір значно змінилась. Багато відомих сайтів використовують MySQL. Наприклад, Facebook, Youtube, Wikipedia.

MySQL - це система управління базами даних, яка використовується для підтримки реляційних баз даних. Це програмне забезпечення з відкритим кодом, що підтримується корпорацією Oracle. Спочатку вона була заснована шведською компанією під назвою MYSQL AB, яку згодом придбали SunMicrosystems і, нарешті, є корпорацією Oracle. Оскільки це база даних із відкритим кодом, вихідний код можна змінювати відповідно до наших потреб.

MySQL – компактний багатопотоковий сервер баз даних. Характеризується високою швидкістю, стійкістю і простотою використання.

MySQL вважається гарним рішенням для малих і середніх застосувань. Сирцеві коди сервера компілюються на багатьох платформах. Найповніше можливості сервера виявляються в UNIX-системах, де є підтримка багатопоточності, що підвищує продуктивність системи в цілому.

Можливості сервера MySQL:

- простота у встановленні та використанні;
- підтримується необмежена кількість користувачів, що одночасно працюють із БД;
- кількість рядків у таблицях може досягати 50 млн;
- висока швидкість виконання команд;
- наявність простої і ефективної системи безпеки.

Переваги:

- Масштабованість. MySQL дає можливості для розробки бази даних як для невеликих проектів, так і для гігантських корпорацій. Функціонал дозволяє легко збільшити або створити додаткові таблиці.
- Підтримка транзакцій. Завдяки підтримці транзакцій гарантується цілісність даних.
- Висока продуктивність. MySQL може бути оптимізованою під будь-які потреби і задовольнити потреби як невеликого блогу, так і великого інтернет магазину.

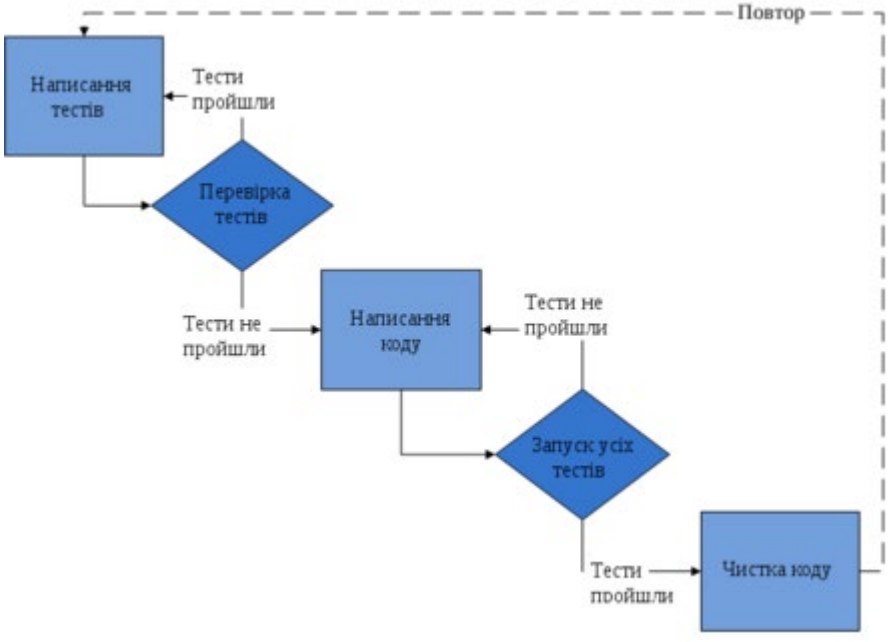
4.3 TDD

Науково-технічний прогрес стимулює прогресивні зміни в усіх сферах діяльності сучасного суспільства. Сучасні технології дозволяють розробляти нові рішення, спрямовані на забезпечення високоякісного рівня життя та підвищення комфорту в користуванні. На сьогоднішній день найактивнішою нішею програмного забезпечення є серверні та веб-додатки. Тому, особливо актуальним питанням в процесі підготовки фахівців з розробки програмного

забезпечення є вибір засобів розгортання та неперервної інтеграції розроблених проектів.

Під час розробки програмного продукту важливу роль відіграють засоби тестування. У світі існують різні методології, які спрямовані на вдосконалення продукту, знаходження помилок ранніх етапах та швидке програмування. Такий підхід як TDD пропонує написання тестів до початку написання робочого коду. Таким чином програміст розбиває задачу на менші завдання і змушує його працювати над однією задачею поки код не пройде всі тести.

Керована тестами розробка (КТР), Розробка через тестування (англ. Test-driven development (TDD)) – технологія розробки програмного забезпечення, яка використовує короткі ітерації розробки, що починаються з попереднього написання тестів, які визначають необхідні покращення або нові функції. Кожна ітерація має на меті розробити код, який пройде ці тести. Нарешті, програміст або група вдосконалюють код для погодження змін. Один із ключових моментів TDD полягає у тому, що підготовка тестів перед написанням самого коду пришвидшує процес внесення змін. Варто зауважити, що керована тестами розробка є методологією розробки програмного забезпечення, а не його тестування.



Переваги:

1. Спрощена, інкрементна розробка. Програміст концентрується на розробці невеликих блоків коду, таким чином просуваючись в розробці. Це краще, ніж заздалегідь виконати велику роботу з проектування і потім виявити в проекті масу неузгодженостей. Також однією з основних переваг TDD є те, що вже на початку розробки одержується робоча система, нехай і з обмеженою функціональністю.
2. Можливість постійного регресійного тестування. В програмуванні часто зустрічається ефект доміно – коли невелика зміна в якійсь частині програми може потягти за собою непередбачувані зміни у функціонуванні всієї системи. Частиною TDD є проведення регресійного тестування після внесення кожної зміни в код, що дає можливість виявляти помилки зразу після їх внесення і таким чином захищає програміста від необхідності високовартісного виправлення помилок в майбутньому.
3. Покращена комунікація і централізація знань. Тести є хорошим методом документування коду – більш точним, ніж словесна або графічна форма і дає можливість розробнику описати ідеї дизайну своєї програми в формі, яка буде зрозумілою іншим.
4. Покращене розуміння вимог до програми, і, відповідно, покращений дизайн програми. Написання тестів до коду дає можливість поглянути на неї спочатку з точки зору користувача і уяснити вимоги до програми краще.
5. Покращена інкапсуляція і модульність. В процесі розробки програміст може внести в код небажані зв’язки між класами. Один з принципів TDD стверджує, що unit-тести повинно бути легко виконати. Це означає, що потрібно мінімізувати вимоги необхідні для запуску тестів. Фокусування на простоті тестів веде до покращення модульності класів, зменшуючи залежності.
6. Зменшення складності за рахунок не внесення надлишкового коду. Розробники часто вносять в код надлишкові методи в сподіванні, що

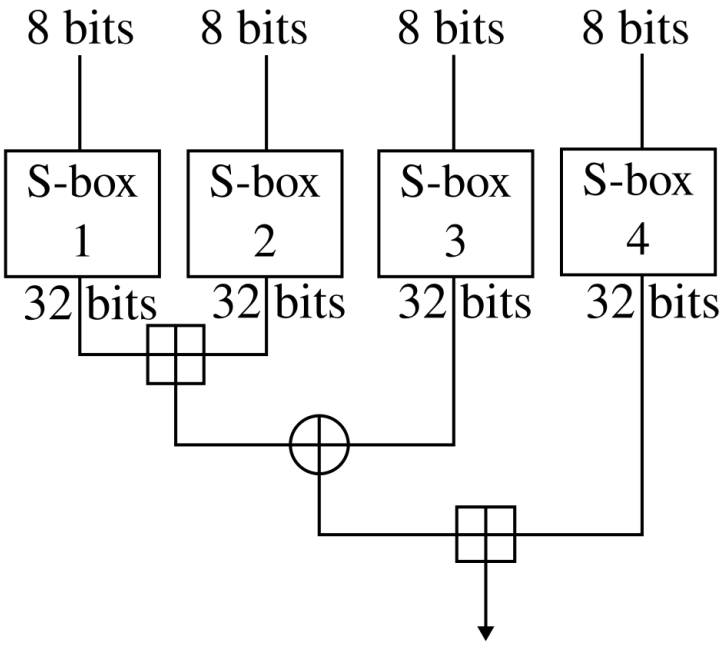
ті потім знадобляться, намагаючись зробити програму більш гнучкою, більш придатною до модифікації в майбутньому. Це веде до збільшення складності програми. Наявність набору тестів дає програмісту більшу впевненість в позитивному результаті внесення змін (навіть досить суттєвих) в код і це приводить до відсутності необхідності в надлишковому коді.

4.4 Шифрування пароля

Для забезпечення доступу у систему використовується ідентифікація користувача за допомогою логіна та пароля. Логін є публічним і може передаватись стороннім особам, але пароль має бути захищеним і схованим. Так як усі запити надсилаються захищеним протоколом https, то залишається найбільш уразливою база даних. Якщо стороння особа отримує доступ до бази даних, то зможе знайти будь-якого користувача і використати його логін і пароль. Для запобігання цьому використовують шифрування пароля і зберігання його у базі даних вже у зашифрованому виді. Щоб отримати пароль треба розкодувати пароль, тому, якщо стороння особа отримала доступ до бази даних користувачів, без знання алгоритму шифрування дані будуть представленні у вигляді набору символів. Знайти і використати корисну інформацію буде неможливо.

Під час розробки веб-системи було використано адаптивну криптографічну хеш-функцію формування ключа bcrypt. Дана функція є адаптивною і може бути легко модифікована для підвищення захисту. Основою шифрування є алгоритм Blowfish. Він реалізує блочне симетричне шифрування зі змінною довжиною ключа.

Алгоритм має дві частини(розширення ключа та шифрування даних). На першому етапі ключ перетворюється в 18 32-бітових підключей і в 4 32-бітних S-блоків, які містять 256 елементів


8 bits
8 bits
8 bits
8 bits
S-box 1
S-box 2
S-box 3
S-box 4
32 bits
32 bits
32 bits
32 bits

.

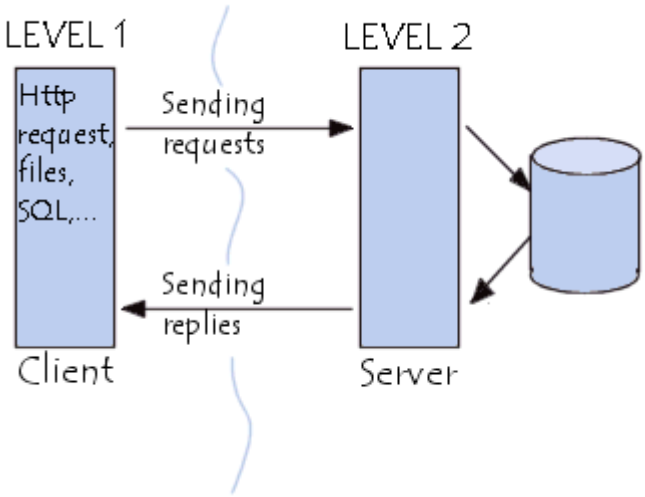
5 РОБОТА КОРИСТУВАЧА З ПРОГРАМНОЮ СИСТЕМОЮ

5.1 Системні вимоги

Для забезпечення коректної роботи потрібен пристрій із стабільним доступом до мережі Інтернет.

5.2 Робота користувача з програмним продуктом

Для роботи програмного продукту повинен бути встановлен браузер на робочому пристрої та доступ до мережі Інтернет. Взаємодія користувача із сервером відбувається виключно через веб-інтерфейс. Далі запит потрапляє на сервер. На сервері запит обробляється і повертаються необхідні дані з бази даних. Таким чином користувач немає прямого доступу до бази даних і вся взаємодія відбувається тільки за допомогою серевра.



6 ВИСНОВКИ

Було проаналізовано вимоги до програмного комплексу, розроблено архітектуру додатку, спроектовано базу даних. На основі отриманих даних було обрано інструменти розробки, мову програмування, вигляд і тип бази даних. Після цього було запрограмовано і протестовано додаток. Практична значущість нашого додатку полягає в можливості підключити його до будь-якої бази даних і отримати зручний інструмент для контролю доступу та відстеження історії звернень до даних.

Додаток 1

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Spring in Action[Електронний ресурс] – Режим доступу до ресурсу: https://www.manning.com/books/spring-in-action-fifth-edition
2. Офіційна документація Spring [Електронний ресурс]. – 2012. – Режим доступу до ресурсу: https://spring.io/docs.
3. Spring Security [Електронний ресурс]. – 2011. – Режим доступу до ресурсу: https://docs.spring.io/spring-security/site/docs/current/reference/htmlsingle/
4. Spring MVC [Електронний ресурс]. – 2010. – Режим доступу до ресурсу: https://docs.spring.io/spring/docs/current/spring-framework-reference/web.html.
5. Introduction to MySQL Database [Електронний ресурс] – Режим доступу до ресурсу: https://www.educba.com/what-is-mysql-database/.
6. The advantages and disadvantages of MySQL [Електронний ресурс] – Режим доступу до ресурсу: http://makble.com/the-advantages-and-disadvantages-of-mysql.
7. Проектирование автоматизированной системы контроля и управления доступом на предприятии [Електронний ресурс] – Режим доступу до ресурсу: https://knowledge.allbest.ru/programming/3c0b65625a3bd69a5d53b89521316c37_0.html.
8. Офіційна документація бази MySQL [Електронний ресурс]. – 2009. – Режим доступу до ресурсу: https://dev.mysql.com/doc/.
9. TDD [Електронний ресурс] 2019 – Режим доступу до ресурсу: https://www.manning.com/books/junit-in-action-third-edition?query=tdd

10. Testing Java [Електронний ресурс] 2018 – Режим доступу до ресурсу: https://www.manning.com/books/testing-java-microservices?query=java
11. Java 8 in Action [Електронний ресурс] 2018 – Режим доступу до ресурсу: https://www.manning.com/books/java-8-in-action?query=java
12. Web Components in Action [Електронний ресурс] 2019 – Режим доступу до ресурсу: https://www.manning.com/books/web-components-in-action?query=web%20archi
13. SQL in Motion [Електронний ресурс] 2017 – Режим доступу до ресурсу: https://www.manning.com/livevideo/sql-in-motion?query=mysql